

lab5 实验报告

一、设计思路

本实验的任务为读代码与翻译，源代码如下：

```
int judge(int r0) {
    int i = 2;
    r1 = 1;
    while (i * i <= r0) {
        if (r0 % i == 0) {
            r1 = 0;
            break;
        }
        i++;
    }
    return r1;
}
```

判断输入的R0是否为质数，要求翻译成如下框架的汇编语言

```
.ORIG x3000
... ; TO BE DONE
HALT
JUDGE ... ; TO BE DONE
... ; TO BE DONE
RET
... ; TO BE DONE
.END
```

step1: 依据该框架对c++进行拆分

在第一个TO BE DONE之前完成一系列初始化问题，对应源代码中的

```
int i = 2;
r1 = 1;
```

在JUDGE一行，即第二个TO BE DONE进行跳转，跳转到另一个子程序，即第四个TO BE DONE执行以下代码功能以下代码

```
i * i <= r0
```

在第三个TO BE DONE 模块对应如下代码

```
if (r0 % i == 0)
{
    r1 = 0;
    break;
}
```

```
i++;
```

step2: 进行关键语句翻译

该程序中有两句翻译比较麻烦，即

```
while(i * i <= r0)
```

```
if (r0 % i == 0)
```

a) 对于第一个语句处理办法如下

由于i的增长是通过i++实现的，而 $(i + 1) * (i + 1)$ 与 $i * i$ 有如下关系

$$(i + 1) * (i + 1) = i * i + 2 * (i + 1) - 1$$

可以用R4 保存每次 $i * i$ 的结果，下一个循环求 $(i + 1) * (i + 1)$ 可以用 $R4 + 2i - 1$ 实现。另外R4初始化为1。

b) 对于第二个语句

通过-R0与i的不断相加来实现，对每次最后的结果进行判断，如果小于0继续减，大于0说明余数不为0，跳转；等于0满足条件，结束子程序。

step3: 全部翻译

翻译代码在下面。

二、实现代码

```
.ORIG x3000
ADD R1 R1 #1;
ADD R2 R2 #2;
NOT R3 R0;
ADD R3 R3 #1; R3 <= -R0
ADD R4 R4 #1;
ST R3 NUM;
JSR JUDGE;
HALT
;
JUDGE BR JUDGE2
label1 ADD R3 R3 R2;
BRn label1;
BRp label2;
BRz label3;
label2 ADD R2 R2 #1;
BR JUDGE;
label3 AND R1 R1 #0;
label4 RET
;
JUDGE2 LD R3 NUM;
ADD R4 R4 R2;
ADD R4 R4 R2;
ADD R4 R4 #-1; 更新 i * i;
ADD R4 R4 R3;
BRp label4;
```

```
LD R3 NUM;  
BR label1  
;  
NUM .BLKW #1  
.END
```

三、性能统计

- 代码行数为26行;
- 平均执行指令数取决于R0, 下面给出一组测试数据

2, 3, 4, 7, 456, 993, 997, 1569, 9293, 121, 9339, 1437 平均执行指令为4200条。